

REMARKS/ARGUMENTS

The Office Action mailed December 28, 2007 has been received and the Examiner's comments carefully reviewed. Claims 1 and 3-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Burkett et al. (U.S. Patent No. 6,635,089) (hereinafter "Burkett") in view of Pik et al. (U.S. Publication No. 2004/0230906) (hereinafter "Pik") and further in view of Messina (U.S. Patent No. 5,634,128). Claim 2 is rejected under 35 U.S.C. 103(a) as being unpatentable over Burkett and Pik and Messina and further in view of Boehme et al. (U.S. Patent No. 6,578,192) (hereinafter "Boehme"). Claims 1, 11 and 20 have been amended. No new matter has been added. The Applicants present the following for consideration.

Claim Rejections

With regard to Claim 1, the Office Action states that Burkett discloses "generating a tree structure that corresponds to the UI script (Figs. 4A&4B and Figs 5A-5E, col. 9, line 44 -col. 11, line 14: the DOM tree (tree structure) is generated and the DOM tree corresponds to the XML document (UI script)); wherein generating the tree structure includes automatically determining templates that were previously grafted to the tree and automatically removing any template that were previously grafted to the tree (Burkett discloses the DOM tree (tree structure) is generated (col. 9, line 11- col. 11, line 14). Burkett also discloses that the DOM API enables application program (automatically) to access a tree-oriented abstraction of a document, and to manipulate document structure and contents (that is, by changing, deleting, and/or adding elements), i.e., when a query-results node is present, the parsed query result is inserted into the DOM tree as a subtree beneath that node, replacing any subtree that previously existed (col. 18, lines 34-38).

Further, the DOM enables navigating the structure of the document (col. 1, lines 35-58). Thus, Burkett discloses generating the tree structure includes automatically determining templates that were previously grafted to the tree and automatically removing any templates that were previously grafted to the tree; ... However, Burkett does not explicitly disclose cloning the reference template to create a cloned reference template while maintaining the reference template, inserting the data into the cloned reference template and grafting the cloned reference template into the tree structure after the data has been inserted into the cloned reference template and displaying a UI output according to the tree structure. ... Messina discloses controlling the access to objects stored in a data processing system wherein the objects are hierarchically structured (tree dom) and each object being either locally available or retrievable from an external unit (col. 2, lines 14-25). Messina further discloses all of the objects contain information about the existence of dependent objects (a counter is associated with each object indicating whether there are objects dependent on the object) (col. 4, lines 1-6). Messina further discloses that through the dependent object counter, a processing program can determine if any locally available object has a dependent object, then appending a dependent object to a locally available object when at least one related dependent object is non-locally available (col. 4, lines 10-15). Messina further discloses that a place-holder graphic symbol (template) is displayed to represent all of the non-locally available dependent objects, and when all of the non-locally available objects are retrieved, the place-holder graphic symbol (template) is removed from the display (tree) and the dependent object is deleted from the tree in the memory (col. 2, lines 34-48). Thus, these steps imply automatically removing any templates that were previously grafted to the tree such that templates that already include data from an external data source are removed

from the tree. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Messina with Burkett and Pik to include automatically removing any templates that were previously grafted to the tree such that templates that already include data from an external data source are removed from the tree for the purpose of improving the performance of applications involving the retrieval of non-locally available objects from external units.” The Applicants respectfully disagree but have amended the Independent Claims to more clearly define the invention.

As amended, Claim 1 recites in part “generating a tree structure that corresponds to the UI script; automatically determining whether the tree structure includes templates that were previously grafted and automatically removing any templates that were previously grafted to the tree such that templates that already include data from an external data source are removed from the tree such that the tree structure is in a state where data binding has yet to occur; wherein the templates are removed before accessing a reference template; accessing the reference template; cloning the reference template to create a cloned reference template while maintaining the reference template such that the reference template is available for subsequent iterations of binding data.” In contrast, the cited references do not teach removing previously grafted branches before accessing a template.

Instead, Burkett at column 18, lines 34-38 teaches that “when a QUERY-RESULTS node is present, the parsed query result is inserted into the DOM tree as a subtree beneath that node (Block 1310), replacing any subtree that previously existed (and which would have represented the results of the earlier query.” Burkett, however, does not teach removing the templates before

the data binding begins. Instead, Burkett teaches replacing a subtree that represented an earlier query. The other cited references do not teach this teaching either. Additionally, none of the cited references teach placing the tree in a state where data binding has not occurred by removing these previously grafted templates. Since the cited references do not teach removing previously grafted templates before the template is accessed, Claim 1 is proposed to be allowable. Claims depending from Claim 1 are proposed to be allowable as they depend on a valid base claim.

As amended, Claim 11 recites in part “generating a tree structure that corresponds to the UI script; automatically determining whether the tree structure includes templates that were previously grafted to the tree structure and automatically removing any templates that were previously grafted to the tree such that templates that already include bound data are removed from the tree; such that the tree structure is in a state where data binding has yet to occur; wherein the templates are removed before cloning a reference template.” For at least the reasons presented above, Claim 11 is proposed to be allowable. Claims 12-19 are proposed to be allowable as they depend from a valid base claim.

As amended, Claim 20 recites in part “generate a tree structure that corresponds to the UI script; automatically determining whether the tree structure includes cloned templates that were previously grafted to the tree structure and automatically removing any cloned templates that were previously grafted to the tree such that templates that already include data from a separate data source are removed from the tree; wherein the previously grafted cloned templates are removed from the tree structure before cloning a reference template.” For at least the reasons

App. No. 10/800,056
Amendment Dated: May 28, 2008
Reply to Final Office Action of December 28, 2007


presented above, Claim 20 is proposed to be allowable. Claims 21-25 are proposed to be allowable as they depend from a valid base claim.

Conclusion

In view of the foregoing amendments, all pending claims are believed to be allowable and the application is in condition for allowance. Therefore, a Notice of Allowance is respectfully requested. Should the Examiner have any further issues regarding this application, the Examiner is requested to contact the undersigned attorney for the applicant at the telephone number provided below.

Respectfully submitted,

MERCHANT & GOULD P.C.



Timothy P. Sullivan
Registration No. 47,981
Direct Dial: 206.342.6254

MERCHANT & GOULD P.C.
P. O. Box 2903
Minneapolis, Minnesota 55402-0903
206.342.6200

